O.K. TO BE ENTERED, July 24, 2009, /BEN C. WANG/

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111
TELEPHONE (801) 533-9800
FAX (801) 328-1707

TELECOPIER COVER SHEET

July 20, 2009

Total Number of Pages
(including cover letter): 14 pages

Please deliver the transmitted facsimile pages to:

Examiner Ben C. Wang

Phone: (571) 270-1240

Fax: (571) 270-2240

From: Brian D. Tucker

Comments: Please see the attached interview issues to be discussed Wednesday, July 22, 2009 at 2:30pm EST

Serial No. 10/822,454

Docket No. 13768.1433

****

We are transmitting from a Sharp FO-750 or Sharp FO-6100 facsimile machine. If you do not receive all the pages or they are unreadable, please contact me as soon as possible at (801) 533-9800.

2453842_1.DOC

BRIAN D. TUCKER
OFFICE DIRECT: 801-322-8427
EMAIL: btucker@WNLAW.com

VIA eFILE

PATENT APPLICATION
Docket No. 13768.1433

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of		)
	Erez Haba	)
Serial No.:	10/822,454	) Art Unit) 2192
Filed:	April 12, 2004	)
Conf. No.:	2824	)
For:	VERSION AWARE TEST MANAGEMENT SYSTEM AND METHOD	)
Examiner:	Ben C. Wang	)
Customer No.:	47973	)

AMENDMENT "F" AND RESPONSE
AFTER FINAL

VIA eFILE AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Final Office Action of April 16, 2009 (paper no. 20090408), please amend the above-identified application as follows:

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

Remarks/Arguments begin on page 9 of this paper.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) An application test management system that maintains fine-grained versioning of tests and their relationship to software under test without sacrificing querying, filtering, and reporting, the system comprising:

a computer readable storage medium having stored thereon the following components executable by a processor:

a version component that ~~monitors~~ detects versions of a source under test ~~components~~ and versions of one or more tests that test the source under test; ~~test components for~~ changes;

a test case file component that ~~includes~~ receives metadata that defines which versions of the one or more tests test which versions of the source under test, and stores the metadata in an XML file in conjunction with test results that are generated by executing the one or more tests on the source under test, wherein metadata is also stored which indicates the version of the one or more tests and the version of the source under test to which the test results correspond, the test case file component further storing attributes in the XML file that enable the querying of the test results; and~~associated with test components and source under test components received from the version component that indicates relationships between versions of source under test components and versions of test cases, the test case file component is continuously modified such that new features are added and/or removed to test changes in the source under test components;~~

a component that uses the attributes of the XML file to transform the XML file utilizing XSLT to enable the querying of the test results based on the version of the source under test and the version of the one or more tests which correspond to the test results.

~~a build drop component that comprises an executable version of the software under test and includes changed data from the version component; and~~

~~a test catalog that provides a repository for a collection of test case files, test cases, test variations, and namespace metadata and is constructed from aggregation of individual test case files which relate to each other in a hierarchical fashion;~~

~~wherein the test case file component generates test results that are tagged with the versions of the source under test components and saved to a data store for historical analysis;~~

~~wherein the test case file component receives version data from the version component and stores the version data to an XML document, the XML document is transformed utilizing XSLT to enable a user to view at least one of exception patterns, trends, productivity, and success rates.~~

2. (Currently Amended) The system of claim 1, wherein the ~~test case file component include~~attributes include s a pointer to the source under test.

3. (Currently Amended) The system of claim 1, wherein the ~~test case file component includes~~attributes include a pointer to requirement for test data.

4. (Currently Amended) The system of claim 1, wherein the ~~test case file component includes~~attributes include a pointer to requirement and/or configuration under test data.

5. (Currently Amended) The system of claim 1, wherein the ~~test case file component includes~~attributes include a pointer to a ~~test case component~~.

6-10. (Canceled)

11. (Currently Amended) The system of claim 1, wherein the XML file is stored in a catalog with other XML files, and wherein the XML file has a hierarchical relationship with at least one of the other XML files. ~~wherein the test case component specified in the test case file component is loaded into the test catalog~~.

12. (Currently Amended) The system of claim 11, wherein the test results are generated by a test execution component that executes the one or more tests ~~case~~ on the ~~software~~source under ~~test and generates test results~~.

13-16. (Canceled)

17. (Currently Amended) A test management methodology comprising:

retrieving metadata that defines a version of source code and a version of one or more test cases that test the source code; ~~regarding test version information in relation to software code version under test;~~

persisting the metadata to an XML file in conjunction with test results that are generated by executing the one or more tests on the source code, wherein metadata is also stored which indicates the version of the one or more tests and the version of the source code to which the test results correspond, the test case file component further storing attributes in the XML file that enable the querying of the test results; and ~~a markup language file versioned with test assets and source code;~~

~~continuously modifying test information such that new features are added and/or removed to test version changes to the software code under test, wherein the file is an XML file;~~

~~generating test results that are tagged with test version information in relation to software code version under test, the test results and test version information are all version tagged data and dependent on the versions of the software code under test; and~~

transforming the XML file utilizing XSLT and the attributes to enable a user to view at least one of exception patterns, trends, productivity, and success rates and management operations including at least one of selection, query, reporting, suit composition, and scheduling.

18. (Currently Amended) The method of claim 17, wherein the metadata that defines the versions of the source code and the one or more tests is ~~version information is~~ retrieved from a version component that monitors changes to the source code ~~versions~~ and the one or more tests ~~versions~~.

19. (Canceled)

20. (Currently Amended) The method of claim 17, wherein the attributes ~~file~~ comprises a pointer to at least one of ~~a source under test~~ the source code, a requirement under test, ~~and~~ or a configuration under test.

21. (Canceled)

22. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 17.

23. (Currently Amended) A testing methodology comprising:

loading a test case in accordance with a test case file stored in a source file;

executing the test case on a source code under test;

generating test results, wherein the test results are version tagged to indicate the relationships between test results, version of the test case, and version of the source code under test;

~~continuously modifying test information such that new features are added and/or removed to test version changes to the source code under test;~~

saving the test results to an XML file, wherein the XML file stores metadata that defines the version of the source code and the version of the test case which were executed to generate the test results, ~~associated with the test cases and source code and~~and wherein the XML file further stores pointers to the version of the source code and the version of the test case; and ~~contains at least one of pointers to the source under test, requirements under test and configuration under test;~~

~~providing a repository for a collection of test case files, test cases, test variations, and namespace metadata, wherein the repository is constructed from aggregation of individual test case files which relate to each other in a hierarchical fashion; and~~

employing XSLT to transform the XML file into an in memory representation of a database that enables the test results to be queried. ~~facilitate management operations including at least one of query, reporting, suite composition and scheduling~~.

24. (Canceled)

25. (Original) The method of claim 23, further comprising publishing the test results to an enterprise data store.

26. (Canceled)

27. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 23.

REMARKS

The Final Office Action, mailed April 16, 2009, considered claims 1-5, 11, 12, 14, 17, 18, 20, 22, 23 and 25-27. Claims 1-5, 11, 12, 14, 17, 18, 20, 22, 23 and 25-27 were rejected under 35 U.S.C. § 103(a) as being unpatentable by Blackwell (U.S. Publication No. 2005/0166094) in view of Mandava (U.S. Patent No. 7,203,928) and further in view of Dimitriadis (NPL Reference, *DOM Test Suite Methodology Report*, National Institute of Standards and Technology, February 2004).

By this response, claims 1-5, 11, 12, 17, 18, 20, and 23 are amended, while claims 14 and 26 are canceled. Claims 1-5, 11, 12, 17, 18, 20, 22, 23, 25, and 27 remain pending of which claims 1, 17, and 23 are independent.

The current amendments have been made to better clarify how an XML document (i.e. the Test Case XML file or TCX file) is used to provide the conflicting benefits of tracking versioning of test cases and source code, and maintaining the ability to query and manage test results. Particularly, the versions of source code and tests which test the source code are stored in the XML file. Also, test results are stored that are also versioned. In other words, the XML file maintains the versioning information that defines which test results were generated by running which versions of the tests cases on which version of the source code. Additionally, the XML file also includes attributes which enable the querying of the test results. These attributes enable the data in the XML file to be transformed using XSLT to enable the querying of the test results based on the version of the source code and the version of the test that generated the test results. Each of the independent claims has been amended similarly.

As is described in the Background, there are two divergent approaches for persisting test results during the lifecycle of software development. One approach utilizes databases which are optimized for querying and reporting. However, this approach does not provide an efficient manner to maintain consistency between the source code under test and the test cases. In fact, the only way to maintain version consistency is to perform a full backup of all development assets at the same time, which, because of the amount of time and overhead required, is performed only at important milestones such as product shipment and beta releases. An alternative approach is to store the tests as source code with granular version control that is consistent with the source code under test. This

approach facilitates version synchronization, but does not allow for querying or reporting. As such, this approach is favored by programmer's working alone, and is not suitable for group development.

The present invention provides a way to bridge the gap between these divergent approaches and goals. This is accomplished by persisting not only the version information of the source code and tests in an XML file, but by persisting attributes that enable the persisted data to be transformed via XSLT into an easily queryable format. Specifically, persisting the metadata in the XML file allows versioning consistent with source, and version-aware references to the source under test. The current amendments emphasize these aspects of the invention. Applicant submits that as amended, the cited art fails to teach or suggest each limitation of the independent claims.

Blackwell, for example, is not directed to persisting version information for source code and tests in a manner that allows for querying and reporting as claimed. In contrast, Blackwell discloses embodiments for tracking dependencies between interrelated system components. For example, when a component is modified in Blackwell, a traceability matrix is accessed to determine "a list of test cases that may need to be run to efficiently test the functioning of the complex software system." ¶ 18. This is distinct from the present invention in that Blackwell determines which test cases should be run when the source code is updated whereas the present invention tracks the versions of the source code and the tests run on the source code so that the versions may be persisted with test results for later querying and reporting. In short, Blackwell addresses tracking interdependencies whereas the present invention addresses persisting results. Although this tracking of interdependencies does involve tracking versions of the components, this is not the same as persisting metadata in an XML file that defines which versions of the source code and the tests were used to generate the test results in a manner that enables the data of the XML file to be transformed via XSLT into a format (e.g. a database) that is queryable. Blackwell makes not mention of persisting test results in conjunction with version metadata in an XML file. For this reason, Blackwell fails to teach or suggest each limitation of the independent claims.

Mandava, on the other hand, is directed to generating uniform test results by embedding reporter codes in the tests. Although Mandava states that results of the execution of the test cases can be stored in an XML file, this is distinct from the present invention. Specifically, a key feature of the present invention is not simply the storing of test results in an XML file, but the storing of the

relationships between the version of the test case and the version of the source code which generated the test results. In contrast, the XML file used in Mandava merely provides an indication of which test case was run and what the results of its execution are (e.g. Pass, Fail, or Did Not Run). Mandava fails, however, to disclose the storage of metadata that "defines which versions of the one or more tests test which versions of the source under test, and stores the metadata in an XML file in conjunction with test results that are generated by executing the one or more tests on the source under test, wherein metadata is also stored which indicates the version of the one or more tests and the version of the source under test to which the test results correspond, the test case file component further storing attributes in the XML file that enable the querying of the test results."

In the current office action, the examiner appears to be in accord with Applicant's remarks above that the Blackwell and Mandava references do not disclose persisting version information as was previously claimed. *See, e.g.* Office Action, Pg. 6. Specifically, to address the limitations directed to storing the version information, the examiner cited Dimitriadis. Applicant submits, however, that the Dimitriadis reference also fails to teach or suggest this aspect. The current amendments are intended to clarify this aspect of how the XML file is used to illustrate the distinctions between the present invention and Dimitriadis as well as the other cited references. For example, Dimitriadis is directed to generating tests for testing DOM components. DOM is an interface for accessing structured documents such as XML. *See* Pg. 5. Applicant believes that the most relevant portion of Dimitriadis is on page 9 where it states that JUnit was used to run the tests in the test framework. However, as the Background of the Specification states, JUnit is an open source framework that "allows tests and source under test to be synchronized but prevents the querying and reporting that are necessary for managing a testing effort across a team of any size," and therefore suffers from one of the problems the present invention seeks to resolve.

Further, to address the limitation of "receive[ing] version data from the version component" that appeared previously in the claim, the examiner cited page 12, second paragraph. This paragraph discusses how values can be extracted from an XML document *during the generation of test cases*, and therefore is unrelated to storing version information of tests and source code in an XML file. The examiner also cited page 15, section 2.5.4 as teaching the storage of attribute information that enables the querying of the test results based on the stored version information. However, this section

describes a table that is generated *before the test cases are ever run*. The table merely provides information regarding what tests exists, which DOM components are tested by the tests, and were the tests reside. The table described does not have any relation to the results of testing the source code, nor the information regarding the versions of the tests and the source code used to generate the test results as required by the claims.

Applicant therefore submits that the combination of cited references fails to teach or suggest each limitation of the independent claims such as: "a test case file component that receives metadata that defines which versions of the one or more tests test which versions of the source under test, and stores the metadata in an XML file in conjunction with test results that are generated by executing the one or more tests on the source under test, wherein metadata is also stored which indicates the version of the one or more tests and the version of the source under test to which the test results correspond, the test case file component further storing attributes in the XML file that enable the querying of the test results," as claimed in combination with the remaining limitations.

In view of the foregoing, Applicant respectfully submits that all the rejections to the independent claims are now moot and that the independent claims are now allowable over the cited art, such that any of the remaining rejections and assertions made, particularly with respect to all of the dependent claims, do not need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice, and particularly with regard to the dependent claims.[1]

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at 801-533-9800.

The Commissioner is hereby authorized to charge payment of any of the following fees that may be applicable to this communication, or credit any overpayment, to Deposit Account No. 23-

[1] Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting any official notice taken. Furthermore, although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

3178: (1) any filing fees required under 37 CFR § 1.16; and/or (2) any patent application and reexamination processing fees under 37 CFR § 1.17; and/or (3) any post issuance fees under 37 CFR § 1.20. In addition, if any additional extension of time is required, which has not otherwise been requested, please consider this a petition therefore and charge any additional fees that may be required to Deposit Account No. 23-3178.

Dated this 16th day of July, 2009.

Respectfully submitted,

/Brian D. Tucker/

RICK D. NYDEGGER
Registration No. 28,651
BRIAN D. TUCKER
Registration No. 61,550
Attorneys for Applicant
Customer No. 47973

RDN:BT:jml
2449399_1.DOC